

RESEARCH ARTICLE

Energy-efficient limit cycle walking in disturbance based on nonlinear model predictive control

Yuta Hanazawa , Haruka Nishinami and Shinichi Sagara

Department of Mechanical and Control Engineering, Kyushu Institute of Technology, Kitakyushu, Japan

Corresponding author: Yuta Hanazawa; Email: hanazawa@cmtf.kyutech.ac.jp

Received: 21 April 2023; **Revised:** 19 September 2023; **Accepted:** 13 October 2023;

First published online: 15 November 2023

Keywords: rimless wheel; walking robot; model predictive control; mobile robots

Abstract

In this study, we present a novel approach to generate limit cycle walking using nonlinear model predictive control (NMPC). Output-zeroing control is now widely used as a control method to limit cycle walking. This control offers strong feedback to the desired trajectory and the generation of energy-efficient and robust limit cycle walking. However, we observed that this method disables the natural dynamics of the robot, leading to problems regarding energy efficiency during walking. This study demonstrates that the energy consumption of walking using the output-zeroing control increases significantly in a disturbed environment. To overcome this limitation, the proposed approach leverages the robot's dynamics using NMPC to achieve energy-efficient walking even in a disturbed environment. We demonstrate the practicality of the proposed method using two different simulation environments.

1. Introduction

Recently, several biped robots have been studied as robots that can move on flat terrain as well as uneven and stepped terrain. Zero moment point (ZMP)-based robots such as ASIMO were once the mainstream [1–4], but more recently, many researchers have turned to limit cycle walking robot that can achieve stable periodic gaits without relying on ZMP [5–11]. Passive dynamic walking proposed by McGeer is a pioneer of limit cycle walking robots [5]. Passive dynamic walkers utilize their dynamics to achieve energy-efficient periodic walking on slopes. Many studies have been conducted on attaching motors and other devices to passive dynamic walkers to enable their walking on a flat terrain. Among these, the method of strong convergence to the desired trajectory using output-zeroing control is widely employed for limit cycle walking generation [8–11].

However, methods employing output-zeroing control have a weakness: while they are energy-efficient for walking with convergence to a limit cycle in an ideal environment, they are less efficient in a realistic environment with disturbance such as an uneven terrain. This is because when the state of a robot deviates from the desired trajectory due to a disturbance, the output-zeroing control cancels out the robot's dynamics and converges it to the desired trajectory with a strong feedback.

To counter this, we propose NMPC to take the advantage of robot dynamics for achieving convergence to the desired trajectory and energy efficiency. Although NMPC is computationally expensive and difficult to implement within a control period, recent advances in computing environments and the algorithms for solving nonlinear optimization at high speeds have resulted in an active research on robots with NMPC [12–17].

Therefore, we propose a method for limiting cycle walking generation based on NMPC. Our method produces walking that is more energy-efficient than the walking based on the conventional output-zeroing control method, as demonstrated through several simulation results. In the numerical simulation, we first present the results of a 2D walking analysis using MATLAB and then that of a 3D walking

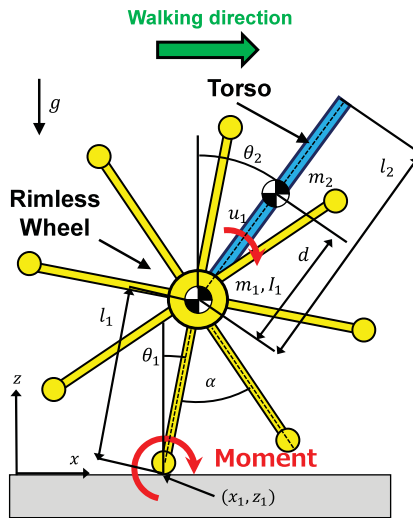


Figure 1. Schematic of the rimless wheel robot with a torso.

analysis using Unity [18]. The robot used in the simulation was made to walk on an uneven terrain, and various disturbances caused by leg collisions with irregular terrain affected the robot’s walking. This paper is organized as follows: Section 2 introduces the rimless wheel robot used in the walking analysis and describes its dynamics model. Next, we discuss the proposed controller based on NMPC and conventional output-zeroing control in Section 3. In Sections 4 and 5, we present the 2D walking analysis results on MATLAB and the 3D walking analysis results on Unity, respectively. Finally, we provide a conclusion and discuss future scope in Section 6.

2. 2D model of the rimless wheel robot

2.1. Equation of motion

Herein, we utilize a rimless wheel robot as a walking robot to simplify the analysis. For the rimless wheel robot, considering the next collision leg as the swing leg results in a walking system similar to that of a biped walking robot [19–24]. The model of the rimless wheel robot is shown in Fig. 1; the robot has a torso and can actuate it using a rotary actuator at the hip joint. The equation of the motion of the rimless wheel robot is expressed as:

$$M_o(q)\ddot{q} + H_{cg}(q, \dot{q}) = Su_1, \tag{1}$$

where $q = [\theta_1, \theta_2]^T$ is the generalized coordinate vector, the inertial matrix $M_o(q) \in \mathbb{R}^{2 \times 2}$ is represented as

$$M_o(q) = \begin{bmatrix} I_1 + (m_1 + m_2)l_1^2 & m_2dl_1\cos(\theta_1 - \theta_2) \\ m_2dl_1\cos(\theta_1 - \theta_2) & m_2d^2 \end{bmatrix},$$

the Coriolis, centrifugal, and gravitational force vectors $H_{cg}(q, \dot{q}) \in \mathbb{R}^2$ are represented as

$$H_{cg}(q, \dot{q}) = \begin{bmatrix} m_2dl_1 \sin(\theta_1 - \theta_2)\dot{\theta}_2^2 - (m_1 + m_2)l_1g \sin \theta_1 \\ -m_2dl_1 \sin(\theta_1 - \theta_2)\dot{\theta}_1^2 - m_2dg \sin \theta_2 \end{bmatrix},$$

u_1 is the input, and the driving matrix $S \in \mathbb{R}^2$ is represented as

$$S = \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

2.2. Impact equation

In the 2D simulation on MATLAB, the collision between a leg and the ground is assumed instantaneous and inelastic. After each collision, the robot’s velocity is calculated using an impact equation [8]. Further, the constraint equation $J_I \in \mathbb{R}^{2 \times 4}$ of the robot when the leg tip touches the ground expressed as:

$$J_I \dot{q}_a = \begin{bmatrix} l_1 \cos \theta_1 - l_1 \cos (\alpha - \theta_1) & 0 & 1 & 0 \\ -l_1 \sin \theta_1 - l_1 \sin (\alpha - \theta_1) & 0 & 0 & 1 \end{bmatrix} \dot{q}_a = \mathbf{0}_{2 \times 1}. \tag{2}$$

The impulse vector $\lambda_I \in \mathbb{R}^2$ and the velocity vector immediately after impact $\dot{q}_a^+ \in \mathbb{R}^4$ are expressed as:

$$\lambda_I = -X_I(q_a)^{-1} J_I \dot{q}_a^-, \tag{3}$$

$$\dot{q}_a^+ = (I_{4 \times 4} - M_a(q_a)^{-1} J_I^T X_I(q_a)^{-1} J_I) \dot{q}_a^-, \tag{4}$$

where

$$X_I(q_a) := J_I M_a(q_a)^{-1} J_I^T, \tag{5}$$

$q_a = [\theta_1, \theta_2, x_1, z_1]^T$ is the augmented generalized variable, x_1 and z_1 denote the position of the stance leg, $\dot{q}_a^- = [\dot{\theta}_1^-, \dot{\theta}_2^-, \dot{x}_1^-, \dot{z}_1^-]^T$ is the velocity immediately before impact, and $M_a(q_a)$ is the following inertia matrix in the augmented generalized variable.

$$M_a(q_a) = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ & M_{22} & M_{23} & M_{24} \\ & & M_{33} & M_{34} \\ Sym. & & & M_{44} \end{bmatrix},$$

where $M_{11} = I_1 + m_2 l_1^2 m_h l_1^2$, $M_{12} = m_2 l_1 a_2 \cos (\theta_1 - \theta_2)$, $M_{13} = (m_2 + m_h) l_1 \cos (\theta_1)$, $M_{14} = -(m_2 + m_h) l_1 \sin (\theta_1)$, $M_{22} = m_2 a_2^2$, $M_{23} = -m_2 a_2 \cos (\theta_2)$, $M_{24} = -m_2 a_2 \sin (\theta_2)$, $M_{33} = m_2 + m_h$, $M_{34} = 0$, $M_{44} = m_2 + m_h$.

3. Controller

3.1. Generating limit cycle walking generation using NMPC

A robot with a torso, such as the rimless wheel robot, can generate propulsive force for walking by maintaining the forward-tilting torso posture. While the desired trajectory of the torso can be a continuous function, herein, we assumed a constant value ($\theta_{2d} = const$). When walking on an uneven terrain, the robot is susceptible to disturbances or state jumps that occur at unpredictable and irregular times intervals; for example, leg-ground collisions, gear backlash, obstacle contact, and leg-floor slippage. In other words, if we can realize a control that converges the torso posture to the desired trajectory with minimal energy consumption under these disturbances, periodically stable and energy-efficient walking can be achieved.

Therefore, NMPC is applied to optimize the convergence of the torso to the desired trajectory and its energy consumption within short prediction intervals because disturbances occur frequently within a gait cycle, making optimization over a long interval impractical. Additionally, short prediction intervals help reduce computational cost and facilitate implementation within a short control cycle of the walking robot. To solve this optimization problem within the control cycle, a fast algorithm is required; we use the C/GMRES method [12], which combines continuous deformation method and GMRES to solve nonlinear optimization problems at a high speed.

3.2. NMPC algorithm

The nonlinear system in a discrete time interval is obtained by

$$x(k+1) = f(x(k), u(k)), \tag{6}$$

$$\mathbf{x}(0) = \mathbf{x}_0, \tag{7}$$

where $\mathbf{x}(k) \in \mathbb{R}^n$ is the state vector and $\mathbf{u}(k) \in \mathbb{R}^m$ is the input vector. Then, we set the following objective function:

$$J := \varphi(\mathbf{x}_N(k)) + \sum_{i=0}^{N-1} L(\mathbf{x}_i(k), \mathbf{u}_i(k)), \tag{8}$$

where

$$\varphi(\mathbf{x}_N(k)) = \frac{1}{2}(\mathbf{x}_N(k) - \mathbf{x}_d(k))^T \mathbf{S}_f(\mathbf{x}_N(k) - \mathbf{x}_d(k)), \tag{9}$$

$$L = \frac{1}{2}(\mathbf{x}_i(k) - \mathbf{x}_d(k))^T \mathbf{Q}(\mathbf{x}_i(k) - \mathbf{x}_d(k)) + \mathbf{u}_i^T(k) \mathbf{R} \mathbf{u}_i(k), \tag{10}$$

$$\mathbf{x}_{i+1}(k) = \mathbf{f}(\mathbf{x}_i(k), \mathbf{u}_i(k)), \tag{11}$$

$$\mathbf{x}_0(k) = \mathbf{x}(k), \tag{12}$$

$i = 0, \dots, N - 1$: N is the number of the predictive steps, \mathbf{x}_d is the desired state vector, \mathbf{S}_f is the terminal weight, \mathbf{Q} is the state weight, \mathbf{R} is the input weight, and $\mathbf{x}_i(k)$ and $\mathbf{u}_i(k)$ are the state and input vectors, respectively, in the prediction interval i -step. The Hamiltonian and optimal necessary conditions are obtained as follows:

$$H(\mathbf{x}_i(k), \mathbf{u}_i(k), \boldsymbol{\lambda}_{i+1}(k)) := L + \boldsymbol{\lambda}_{i+1}^T(k) \mathbf{f}(\mathbf{x}_i(k), \mathbf{u}_i(k)), \tag{13}$$

$$\boldsymbol{\lambda}_N(k) = \frac{\partial \varphi^T}{\partial \mathbf{x}}(\mathbf{x}_N(k)), \tag{14}$$

$$\boldsymbol{\lambda}_i(k) = \frac{\partial H^T}{\partial \mathbf{x}}(\mathbf{x}_i(k), \mathbf{u}_i(k), \boldsymbol{\lambda}_{i+1}(k)), \tag{15}$$

$$\frac{\partial H}{\partial \mathbf{u}}(\mathbf{x}_i(k), \mathbf{u}_i(k), \boldsymbol{\lambda}_{i+1}(k)) = 0, \tag{16}$$

where $\boldsymbol{\lambda}_i(k) \in \mathbb{R}^n$ is the companion variable vector at i -step. The optimal control input is set as the actual input for the system, and then, the finite horizon is moved backward by a discrete interval. This sequence is repeated in the NMPC approach. In C/GMRES, $\mathbf{U}(k) \in \mathbb{R}^{mN}$ and $F(\mathbf{U}(k), \mathbf{x}(k), k) \in \mathbb{R}^{mN}$ are defined by

$$\mathbf{U}(k) := [\mathbf{u}_0^T(k) \cdots \mathbf{u}_{N-1}^T(k)]^T, \tag{17}$$

$$F(\mathbf{U}(k), \mathbf{x}(k), k) := \begin{bmatrix} \mathbf{H}_u^T(\mathbf{x}_0(k), \mathbf{u}_0(k), \boldsymbol{\lambda}_1(k)) \\ \vdots \\ \mathbf{H}_u^T(\mathbf{x}_{N-1}(k), \mathbf{u}_{N-1}(k), \boldsymbol{\lambda}_N(k)) \end{bmatrix}, \tag{18}$$

where $H_u = \frac{\partial H}{\partial \mathbf{u}}$. The state and companion variable vectors can be obtained using Eqs. (13)–(16). Then, $\Delta \mathbf{U}(k)$ is obtained when $\mathbf{F} = 0$ is solved using the C/GMRES algorithm. Finally, $\mathbf{u}(k + 1) = \mathbf{u}(k) + \Delta T \Delta \mathbf{u}(k)$ can be calculated.

Moreover, we introduce an integrator to suppress the oscillation of the control input as the input is unstable when there are disturbances or state jumps [13–15]. To achieve this, we designed an augmented

system that includes the input as a part of the state. The augmented system for the rimless wheel robot is expressed as follows:

$$\begin{aligned} \mathbf{x}(k+1) &= \begin{bmatrix} y_1(k+1) \\ y_2(k+1) \\ y_3(k+1) \\ y_4(k+1) \\ u_1(k+1) \end{bmatrix} = \begin{bmatrix} y_1(k) + y_3(k)\Delta T \\ y_2(k) + y_4(k)\Delta T \\ y_3(k) + \ddot{\theta}_1\Delta T \\ y_4(k) + \dot{\theta}_2\Delta T \\ u_1(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \Delta T \end{bmatrix} \Delta u_1, \\ &= \mathbf{f}(\mathbf{x}(k)) + \mathbf{g}\Delta u_1, \end{aligned} \tag{19}$$

where

$$\begin{bmatrix} \ddot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \mathbf{M}_o^{-1}(\mathbf{q}(k))(\mathbf{S}u_1(k) - \mathbf{H}_{cg}(\mathbf{q}(k), \dot{\mathbf{q}}(k))), \tag{20}$$

$y_1(k)$ is the output of the angle of the stance leg, $y_2(k)$ is the output of the angle of the torso, $y_3(k)$ is the output of the angular velocity of the stance leg, $y_4(k)$ is the output of the angular velocity of the torso, ΔT is the discrete interval, and Δu_1 is the differential value of the input. The objective function is defined as follows:

$$J := \varphi(\mathbf{x}(N), \mathbf{y}_d(k)) + \sum_{i=0}^{N-1} L(\mathbf{x}(k), \Delta \mathbf{u}(k)), \tag{21}$$

where

$$\varphi(\mathbf{x}(N), \mathbf{y}_d(k)) = \frac{1}{2}(\mathbf{y}(N) - \mathbf{y}_d(N))^T \mathbf{S}_f (\mathbf{y}(N) - \mathbf{y}_d(N)), \tag{22}$$

$$\begin{aligned} L(\mathbf{x}_k, \Delta u_1(k)) &= \frac{1}{2}(\mathbf{y}(k) - \mathbf{y}_d(k))^T \mathbf{Q}(\mathbf{y}(k) - \mathbf{y}_d(k)) \\ &\quad + \frac{1}{2}R\Delta u_1^2(k), \end{aligned} \tag{23}$$

$\mathbf{y}_d(k)$ is the desired output vector. Then, the Hamiltonian condition is defined as

$$H(\mathbf{x}(k), u_1(k), \boldsymbol{\lambda}(k+1)) := L + \boldsymbol{\lambda}^T(k+1)(\mathbf{f}(\mathbf{x}(k)) + \mathbf{g}\Delta u_1). \tag{24}$$

From Eq. (24), the optimal necessary condition is obtained by

$$\frac{\partial H}{\partial \Delta u} = \Delta u_1^T R + \boldsymbol{\lambda}^T(k+1)\mathbf{g}. \tag{25}$$

$$= 0. \tag{26}$$

The optimal control input, Δu_1 , can be calculated, and u_1 is obtained by integrating Δu_1 .

3.3. Output-zeroing control

For comparison with the proposed method, a conventional method of output-zeroing control is introduced here [10]. This method is input-output linearization and output-zeroing control for the torso’s attitude control. Although the actual robot is 3D, the mathematical model for the control is defined as 2D since the rimless wheel robot performs 2D motion constrained to the sagittal plane. The affine system is defined as:

$$\frac{d}{dt} \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} := \mathbf{f}_a(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}_a(\mathbf{q})u_1. \tag{27}$$

$$\mathbf{f}_a(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{M}_o^{-1}(\mathbf{q})(-\mathbf{H}_{cg}(\mathbf{q}, \dot{\mathbf{q}})) \end{bmatrix}$$

$$\mathbf{g}_a(\mathbf{q}) = \begin{bmatrix} 0 \\ \mathbf{M}_o^{-1}(\mathbf{q})\mathbf{S} \end{bmatrix}$$

To converge the torso angle θ_2 to the desired torso angle θ_{2d} , the output function is defined as

$$y := h(q) = \theta_2 - \theta_{2d} = 0. \tag{28}$$

Differentiating this function with time yields the following functions.

$$\begin{aligned} \frac{dy}{dt} &= \begin{bmatrix} \frac{\partial h}{\partial q} & 0 \end{bmatrix} \begin{bmatrix} \dot{q} \\ \mathbf{M}_o^{-1}(q) (-\mathbf{H}_{cg}(q, \dot{q})) \end{bmatrix} \\ &\quad + \begin{bmatrix} \frac{\partial h}{\partial q} & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{M}_o^{-1}(q) \mathbf{S} \end{bmatrix} u_1 \\ &= L_f h + L_g h, \end{aligned} \tag{29}$$

$$\begin{aligned} \frac{d^2y}{dt^2} &= \begin{bmatrix} \frac{\partial}{\partial q} (\frac{\partial h}{\partial q} \dot{q}) & \frac{\partial h}{\partial q} \end{bmatrix} \begin{bmatrix} \dot{q} \\ \mathbf{M}_o^{-1}(q) (-\mathbf{H}_{cg}(q, \dot{q})) \end{bmatrix} \\ &\quad + \frac{\partial h}{\partial q} \mathbf{M}_o^{-1}(q) \mathbf{S} u_1 \\ &= L_f^2 h + L_g L_f h u_1. \end{aligned} \tag{30}$$

Since the relative order of the system is two, the partially linearized feedback input for input-output linearization is given by

$$u_1 = -\frac{1}{L_g L_f h} (L_f^2 h - v), \tag{31}$$

where $L_g L_f h$ and $L_f^2 h$ are given by

$$\begin{aligned} L_g L_f h &= \mathbf{C} \mathbf{M}_o^{-1}(q) \mathbf{S}, \\ L_f^2 h &= -\mathbf{C} \mathbf{M}_o^{-1}(q) \mathbf{H}_{cg}(q, \dot{q}), \\ \mathbf{C} &= [0 \ 1]. \end{aligned}$$

Finally, we set the new control input v as

$$v = -K_p(\theta_2 - \theta_{2d}) - K_d \dot{\theta}_2. \tag{32}$$

4. 2D walking analysis in MATLAB

4.1. Setting for simulation

We conducted a 2D walking simulation using MATLAB to compare the performance of NMPC and output-zeroing control. This simulation assumed level ground, fully inelastic collisions, and no slippage, while disturbances were applied to each joint (the contact point of the support leg and the hip joint) as normally distributed random torques. Tables I, II, and III show physical and control parameters on MATLAB simulations. We set the initial state as $\mathbf{X}_0 = [-0.20 \ 0.35 \ 1.50 \ 0.00]^T$, and The desired values as $\theta_{2d} = 0.35$ rad and $\dot{\theta}_{2d} = 0$ rad/s. Furthermore, we analyzed the walking speed and the energy efficiency of walking (20 steps) using NMPC and output-zeroing control. The specific resistance (SR) metric was used to evaluate the energy efficiency, which is given by:

$$SR := \frac{P}{M g v_a}, \tag{33}$$

where p [J/s] is the average input power, M [kg] is the total weight of the robot, and v_a [m/s] is the average walking speed. The average input power is defined as

$$p := \frac{1}{T} \int_0^T |u_1(\dot{\theta}_2 - \dot{\theta}_1)| dt,$$

where T [s] is the total walking time.

Table I. Physical parameters in the MATLAB simulations.

Symbol	Value	Unit 3
m_1	2.0	kg
m_2	1.0	kg
l_1	0.4	m
l_2	0.6	m
d	0.3	m
I_1	0.16	kg · m ²
g	9.8	m/s ²
α	$\pi/4$	rad

Table II. Control parameters of output-zeroing control in the MATLAB simulations.

Symbol	Value	Unit
K_p	5000	–
K_d	90	–

Table III. Control parameters of NMPC in the MATLAB simulations.

Symbol		Value	Unit
S_f	Terminal weight	diag[0, 15,000, 0, 200, 0]	–
q	State weight	diag[0, 13,000, 0, 150, 0]	–
R	Input weight	0.045	–
T_f	Length of the final evaluation interval	0.05	s
α	Parameter of the evaluation interval	0.9	–
N	Number of divisions in the evaluation interval	5	–
Δt	Sampling period	0.005	s
ζ	Parameter for continuous deformation method	$1/\Delta t$	–
k_{max}	Maximum number of iterations	2	–
δt	Time difference for forward differential approximation	0.001	s
er	Allowable limit of error	1.0×10^{-8}	–

4.2. Parameter setting for NMPC

The key points for determining the parameters in Table III are S_f , Q , R , T_f , and N . First, T_f is the length of the prediction interval and is set to a short value as demonstrated in Section 3.1. Various disturbances can occur at unexpected times during the robot’s gait. When the robot deviates from the target trajectory because of such disturbances, we aimed to realize an entrainment that quickly and efficiently pulls the robot’s state back to the target trajectory using NMPC. Therefore, herein, the generation of optimal entrainment to target trajectory by NMPC is main idea. In this study, $T_f = 0.05$ was set by trial and error and N is the number of divisions of this prediction interval, T_f . Herein, this number of divisions is set to $N = 5$ in terms of computation time. If a shorter computation time is desired, we have set $N = 3$ or 5 ($N = 3$ in Section 5). For S_f , Q , R a trial-and-error approach is used to find values for which the robot state converges sufficiently to the target values. From there, the input weight R is increased to evaluate the values that improve energy efficiency. Other parameters are determined with reference to previous studies [12].

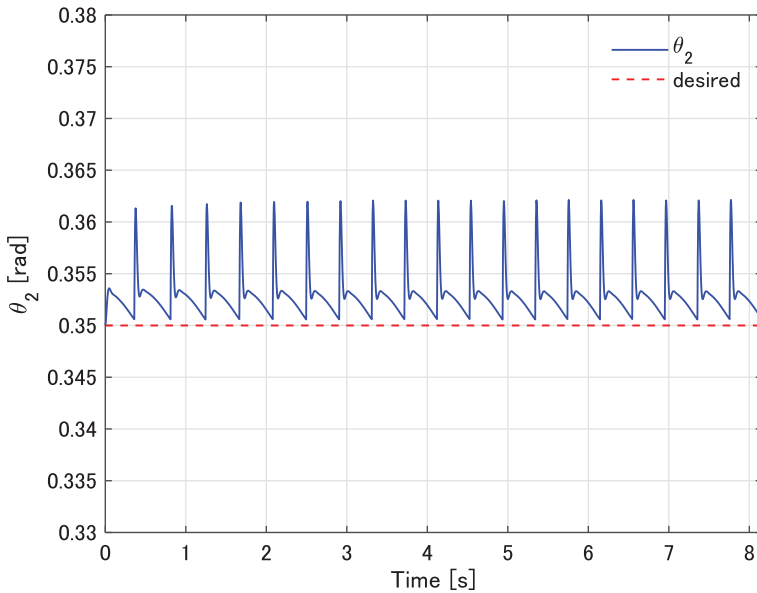


Figure 2. Torso angle with respect to time.

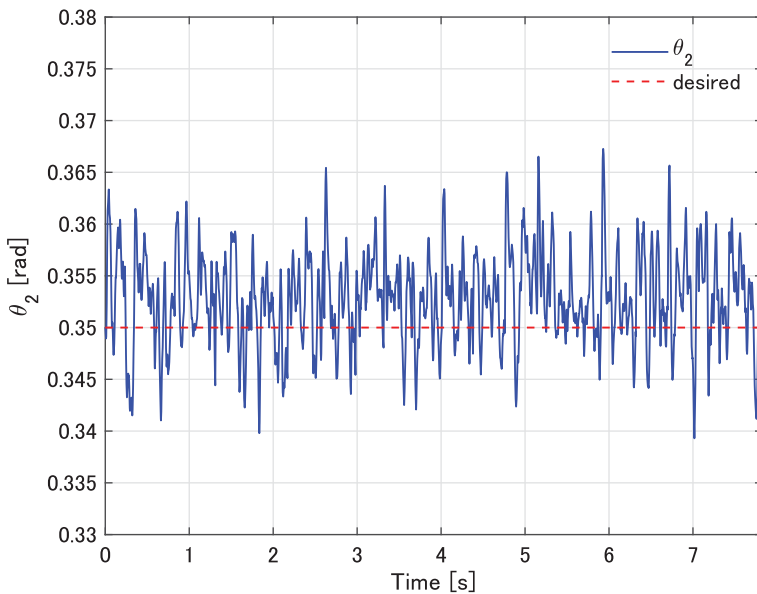


Figure 3. Torso angle in disturbance with respect to time.

4.3. Walking using output-zeroing control

First, we analyzed the walking using the output-zeroing control. Figures 2 and 3 show the torso angle in walking without and with disturbance, respectively. The robot achieved periodic walking using output-zeroing control without and with disturbance, the walking speed was 0.757 m/s, the SR was 0.071 without disturbance, the walking speed was 0.806 m/s, and the SR was 0.158 with disturbance. The nature of the given disturbance caused a slight increase in the walking speed. Further, we confirmed that the disturbance greatly reduced the energy efficiency of walking.

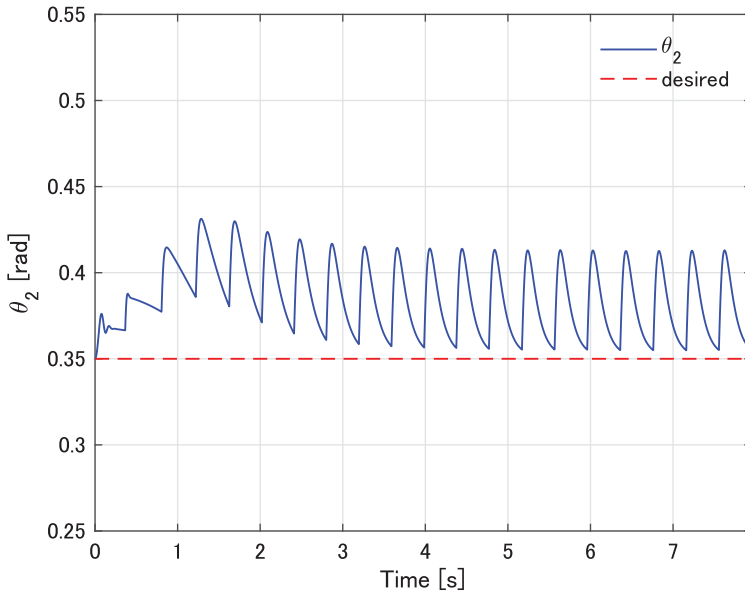


Figure 4. Torso angle with respect to time.

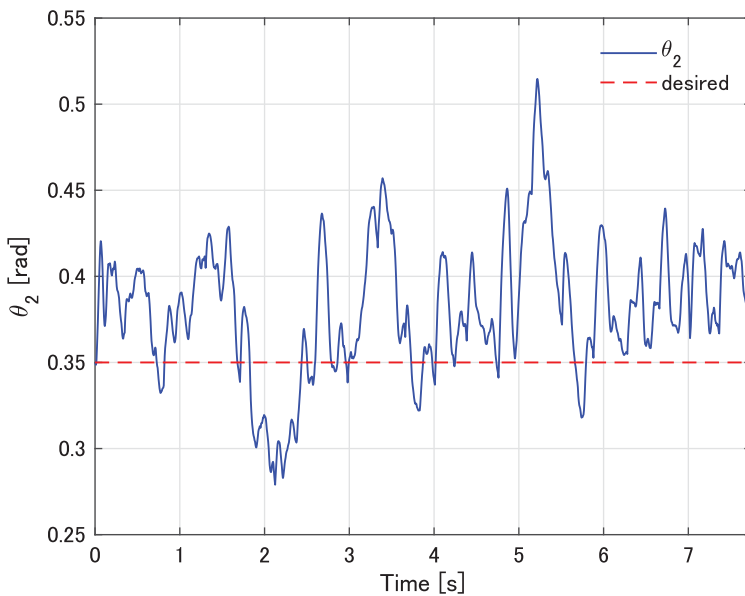


Figure 5. Torso angle with respect to time in a disturbance.

4.4. Walking using NMPC

We analyzed the walking performance using NMPC under the same conditions as the output-zeroing control by including the same disturbances. Figures 4 and 5 show the torso angle in walking without and with disturbance, respectively. The robot achieved periodic walking using NMPC without and with disturbance, the walking speed was 0.772 m/s, the SR was 0.076 without disturbance, the walking speed was 0.824 m/s, and the SR was 0.084 with disturbance. The walking speed using NMPC was almost the same as observed in output-zeroing control. However, the energy efficiency was

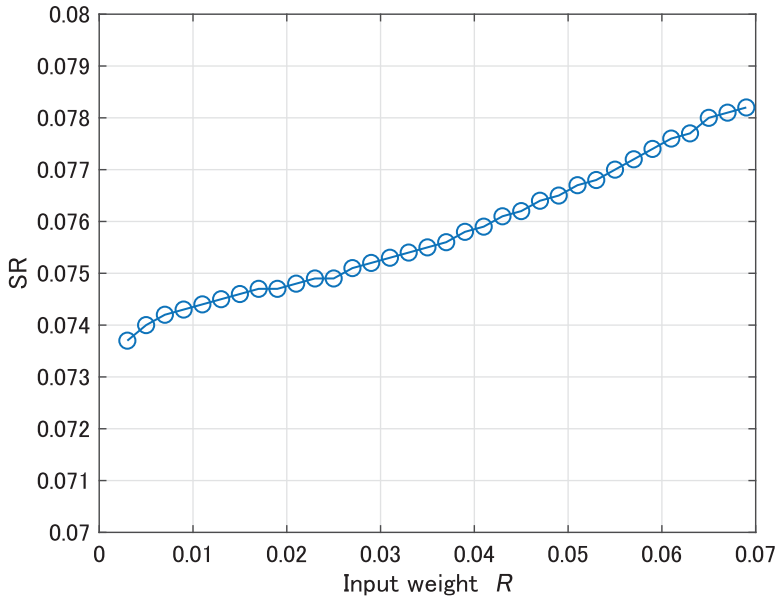


Figure 6. SR with respect to the input weight of NMPC.

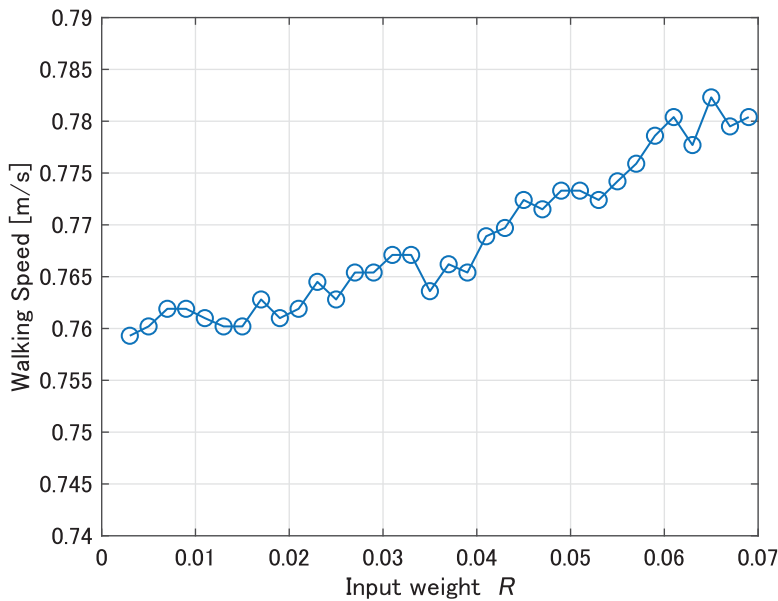


Figure 7. Walking speed with respect to the input weight of NMPC.

improved by nearly double. This can be attributed NMPC optimizing the energy consumption under the disturbances.

Then, we analyzed the SR values and walking speeds for the input weight of NMPC. Figures 6 and 7 show SR values and walking speeds with respect to the input weights of NMPC, respectively. We observed that SR values and walking speeds increase slightly with respect to the input weight. Further, Figs. 8 and 9 show SR values and walking speeds with respect to the input weight of NMPC in disturbance, respectively. These figures indicate that SR monotonically decreases with increasing input weight, while the walking speed is monotonically increases with the increasing input weight.

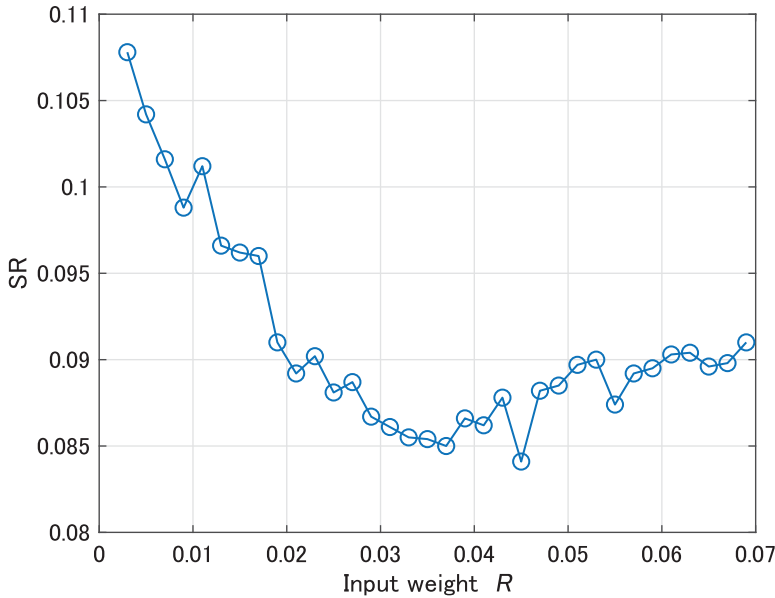


Figure 8. SR with respect to the input weight of NMPC in disturbance.

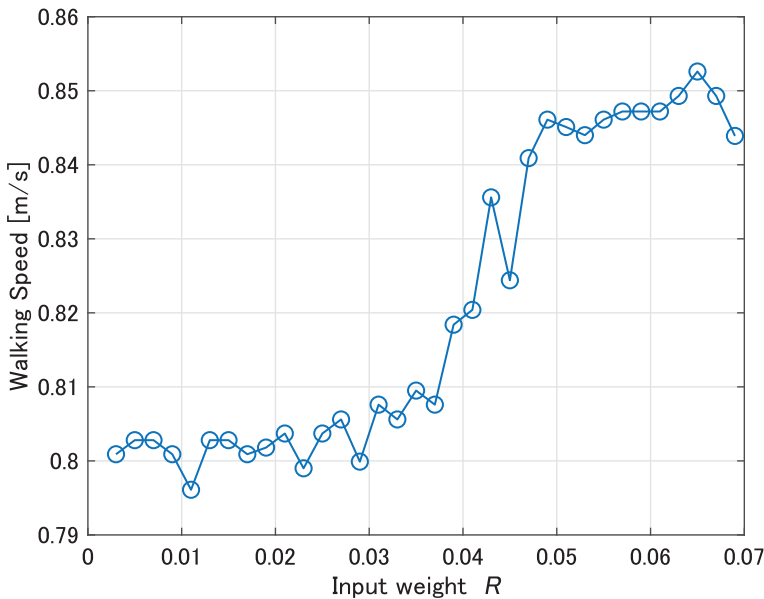


Figure 9. Walking speed with respect to the input weight of NMPC in disturbance.

The increase in walking speed is attributed to the nature of the disturbance; by increasing the input weights, the motion was designed to take advantage of the disturbance, which increased the walking speed and further optimized energy consumption. Table IV and Fig. 10 summarize the results of the above analysis. We observed no significant difference in the walking speed of output-zeroing control and NMPC under disturbances. However, there was a significant difference in their energy consumption

Table IV. Comparison of the SR values and average walking speeds.

	SR	Speed [m/s]
Output-zeroing control	0.071	0.757
Output-zeroing control in disturbance	0.158	0.806
Proposed NMPC	0.073–0.078	0.759–0.782
Proposed NMPC in disturbance	0.084–0.107	0.796–0.852

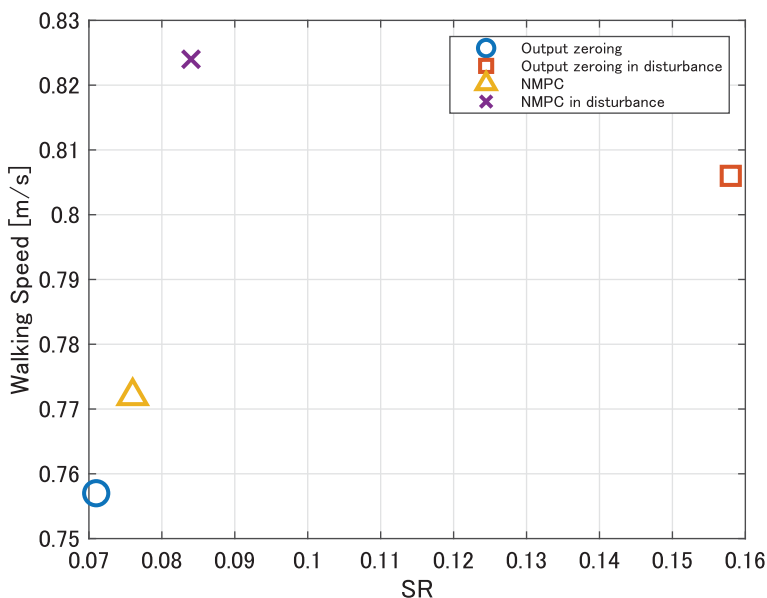


Figure 10. Comparison of the SR values and average walking speeds ($R = 0.045$).

in the presence of a disturbance. The NMPC produced energy-efficient walking even in the presence of disturbance.

5. 3D walking analysis in Unity

5.1. Simulation setting

To perform the walking analysis in an environment that closely resembles an actual environment, we carried out simulations in a 3D environment using Unity. We demonstrate the practicality of walking with NMPC through an uneven surface as shown in Fig. 11. This uneven surface was created using Terrain Toolkit 2017, with the environment set to “PATH OF THE FLESH” and the maximum ground altitude was set to 0.6 m. The simulation communicated between Unity and MATLAB and the control input was computed by the same MATLAB program, as shown in Fig. 12. Table V shows the physical parameters of the robot in Unity. These parameters are those of our previous robots [22], and the robot inertia, I_2 , around the center of gravity of the body since the body is rectangular. Therefore, the inertia matrix, M_o , is modified as follows:

$$M_o(q) = \begin{bmatrix} I_1 + (m_1 + m_2)l_1^2 & m_2dl_1\cos(\theta_1 - \theta_2) \\ m_2dl_1\cos(\theta_1 - \theta_2) & I_2 + m_2d^2 \end{bmatrix}.$$

Table V. Physical parameters of the robot in the Unity simulations.

Symbol	Value	Unit
m_1	0.365	kg
m_2	0.450	kg
l_1	0.180	m
l_2	0.174	m
d	0.085	m
I_1	$4.07 e^{-3}$	$\text{kg} \cdot \text{m}^2$
I_2	$2.0 e^{-3}$	$\text{kg} \cdot \text{m}^2$
g	9.8	m/s^2
α	$\pi/4$	rad

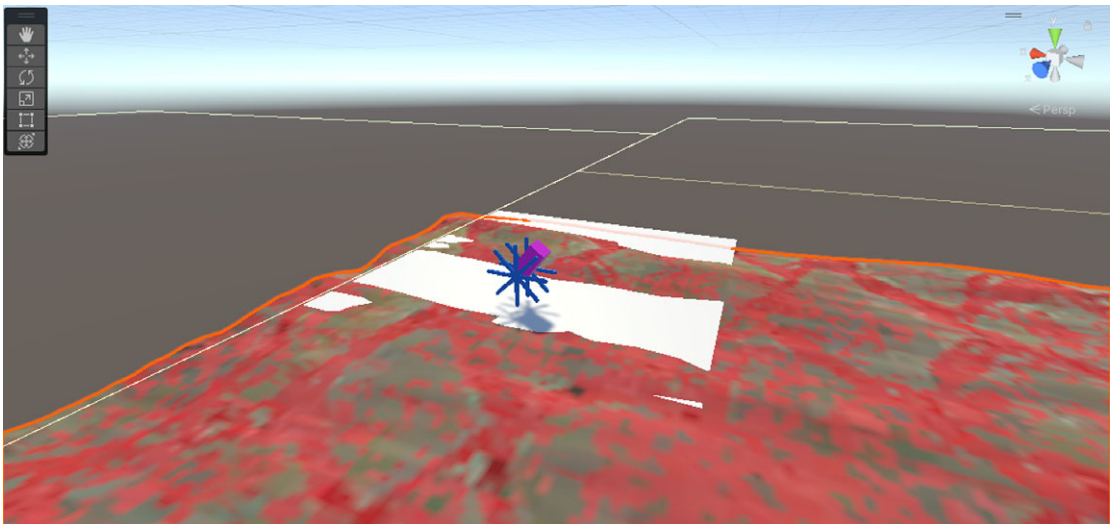


Figure 11. Screenshot of the walking environment in Unity.

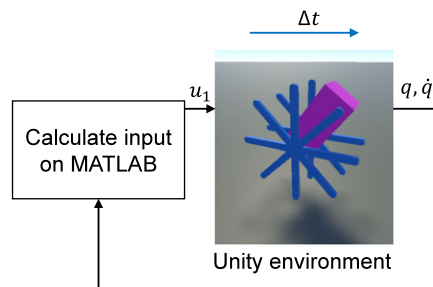


Figure 12. Overview of the connection between MATLAB and Unity.

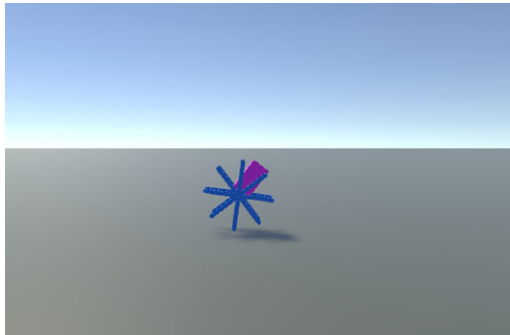
Tables VI and VII show the control parameters of the output-zeroing control and NMPC in Unity, respectively. The desired values are $\theta_{2d} = 0.65$ rad and $\dot{\theta}_{2d} = 0$ rad/s, and the initial state is $\mathbf{x}_0 = [0.00 \ 0.65 \ 8.0 \ 0.00]^T$.

Table VI. Control parameters of output-zeroing control in the Unity simulations.

Symbol	Value	Unit
K_p	30,000	–
K_d	150	–
θ_{2d}	0.65	rad

Table VII. Control parameters of NMPC in the Unity simulations.

Symbol	Value	Unit
S_f	diag[0, 28,000, 0, 4, 0]	–
q	diag[0, 26,000, 0, 4, 0]	–
R	1.0	–
T_f	0.05	s
α	0.9	–
N	3	–
Δt	0.01	s
ζ	$1/\Delta t$	–
k_{\max}	2	–
δt	0.001	s
er	1.0 e^{-8}	–

**Figure 13.** Walking on the flat surface.

Figures 13 and 14 show the robot's walking on a flat surface and on the uneven terrain for 20 s, respectively. Here, three types of uneven terrains (no. 1, 2, and 3) were created with random shapes, and a walking analysis was performed.

5.2. Walking by output-zeroing control in Unity

First, we analyze the walking using the output-zeroing control on the flat surface and the uneven terrain. Figures 15 and 16 show the torso angle during walking on a flat surface and on an uneven terrain no. 1, respectively. These figures show the robot performs periodic walking on the flat surface and on the uneven terrain. Table VIII and Fig. 17 show the SR values and the average speeds of walking. It was observed that walking on a flat surface is the most energy-efficient and the fastest. However, the walking speed and energy efficiency were reduced in all uneven terrains.

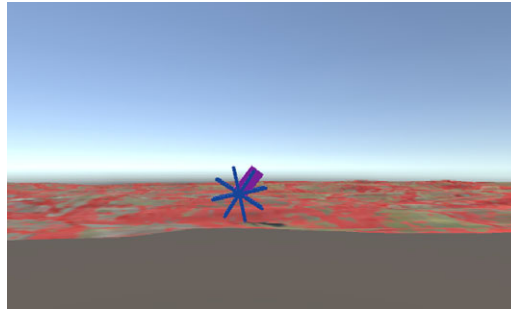


Figure 14. Walking on the uneven terrain.

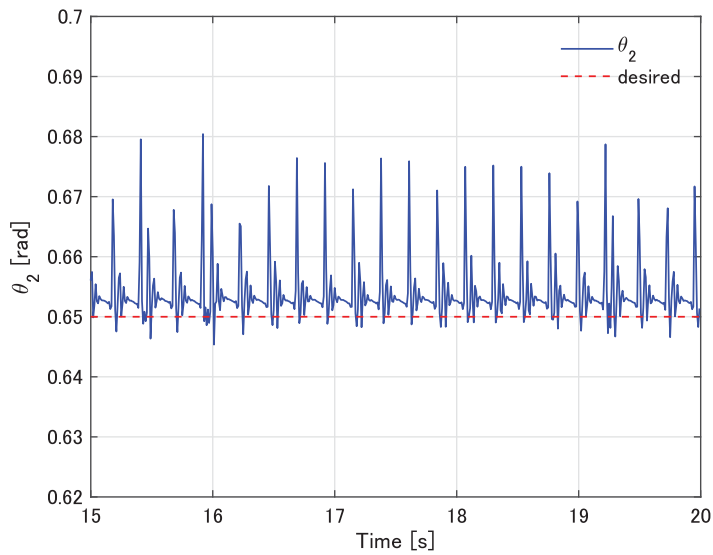


Figure 15. Torso angle in the walking using the output-zeroing control on the flat surface.

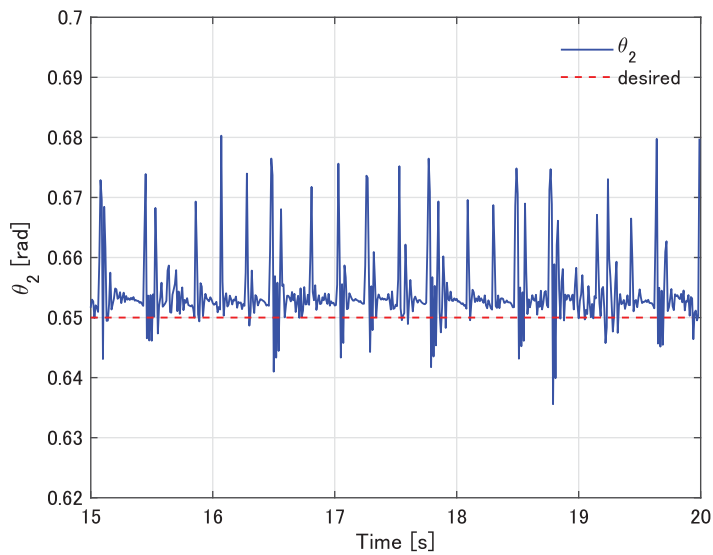
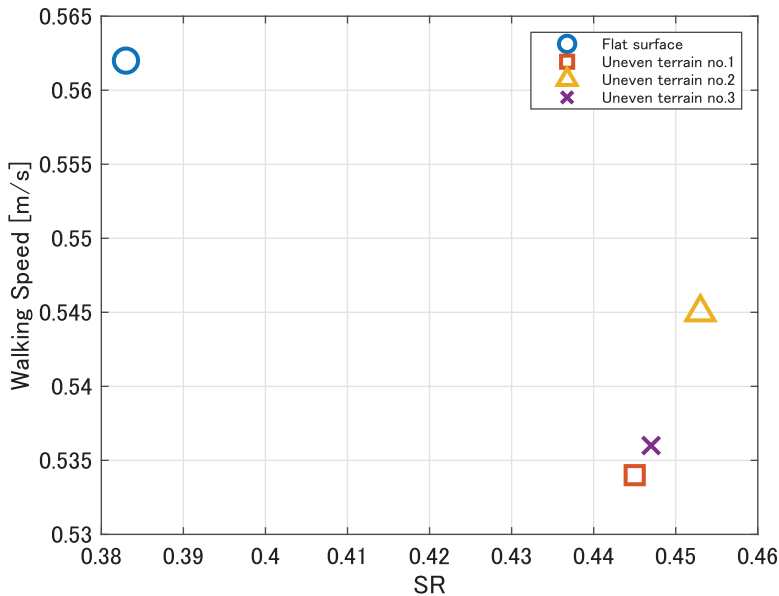


Figure 16. Torso angle in the walking using the output-zeroing control on the uneven terrain no. 1.

Table VIII. Comparison of the SR values and average walking speeds using output-zeroing control in Unity.

	SR	Speed [m/s]
Flat surface	0.383	0.562
Uneven terrain no.1	0.445	0.534
Uneven terrain no.2	0.453	0.545
Uneven terrain no.3	0.447	0.536

**Figure 17.** Comparison of the SR values and average walking speeds using output-zeroing control in Unity.

5.3. Walking using NMPC in Unity

Next, we analyzed walking using NMPC in Unity in the same conditions. Figures 18 and 19 show the torso angle in walking using NMPC on the flat surface and the uneven terrain no. 1, respectively. As with output-zeroing control, we observed that the robot can periodically walk on a flat surface and uneven terrain. Table IX and Fig. 20 show the SR values and average speeds for walking using NMPC. Unlike the output-zeroing control, the energy efficiency did not almost decrease while walking on uneven terrains.

Further, we analyzed the walking with varying input weights of NMPC. Figures 21 and 22 show the SR values of walking using the NMPC on the flat surface and on the uneven terrain no. 1 with respect to the input weight, respectively. Similarly, Figs. 23 and 24 show the walking speeds of the NMPC walking on the flat surface and on the uneven terrain no. 1 with respect to the input weight, respectively. We observed that, as the input weight increased, the walking performance improved. In this case, setting the input weight to a value >1 enabled energy-efficient walking. Finally, we compared the SR values and walking speeds obtained for the output-zeroing control and NMPC in Table X and Fig. 25. Comparing the output-zeroing control and NMPC on a flat surface and on the uneven terrain, it is found that NMPC outperforms the output-zeroing control in energy efficiency and walking speed. Therefore, it can be concluded that NMPC is upward compatible with output-zeroing control in this simulation case.

Table IX. Comparison of the SR values and average walking speeds using NMPC in Unity.

	SR	Speed [m/s]
Flat surface	0.266	0.658
Uneven terrain no.1	0.268	0.693
Uneven terrain no.2	0.267	0.679
Uneven terrain no.3	0.269	0.677

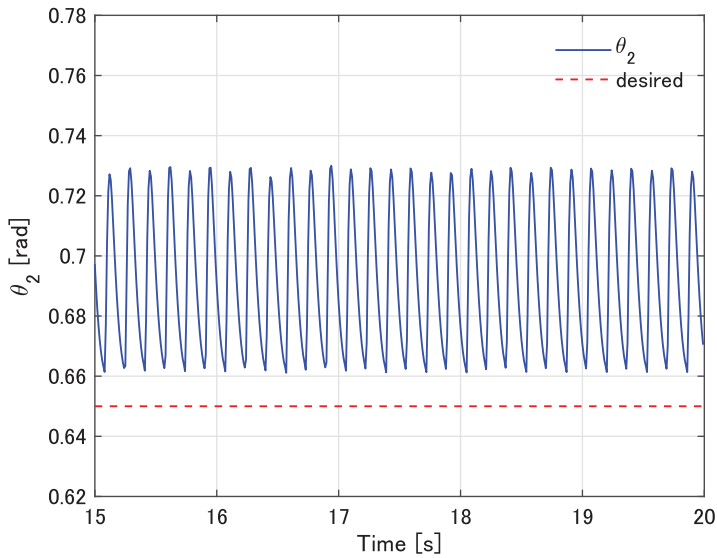


Figure 18. Torso angle in the walking using NMPC on a flat surface.

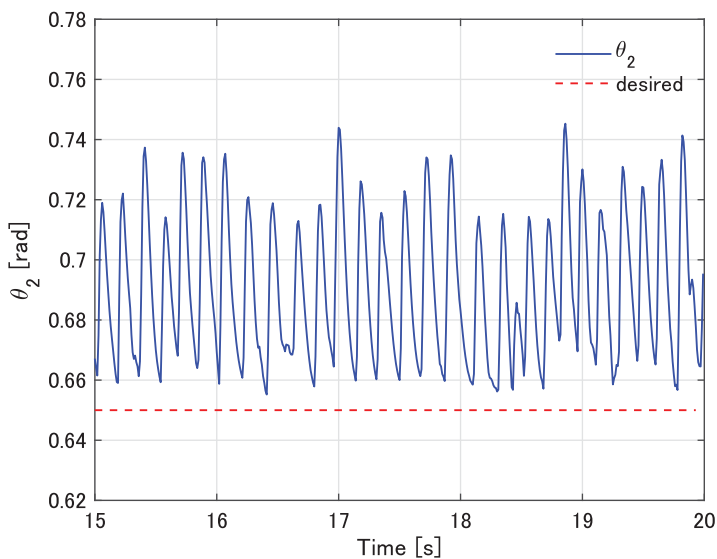


Figure 19. Torso angle in the walking using NMPC on the uneven terrain no. 1.

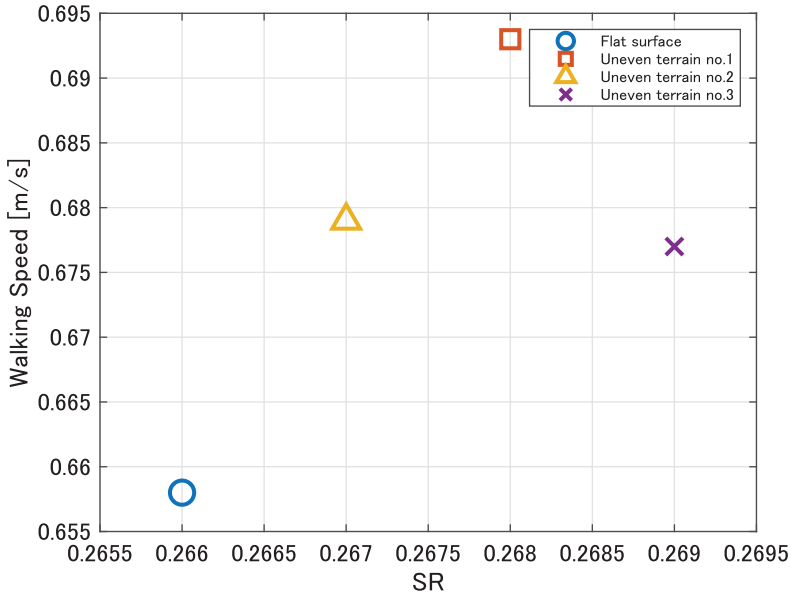


Figure 20. Comparison of the SR values and average walking speeds using NMPC in Unity.

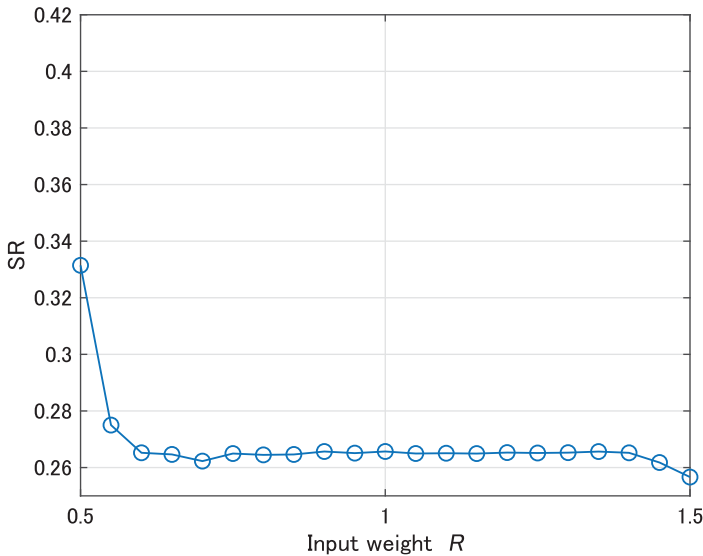


Figure 21. SR of the walking using NMPC on the flat surface with respect to the input weight (R).

6. Conclusion and future works

This study proposes an NMPC-based method for achieving energy-efficient limit cycle walking in a disturbed environment. We first developed a 2D simulator using MATLAB and performed analyses under the assumption of a strictly perfect inelastic collision, which is common in walking analysis. Then, a 3D walking simulator was developed using Unity to verify the practicality of NMPC under uneven terrain conditions. In both environments, walking with the proposed method showed excellent

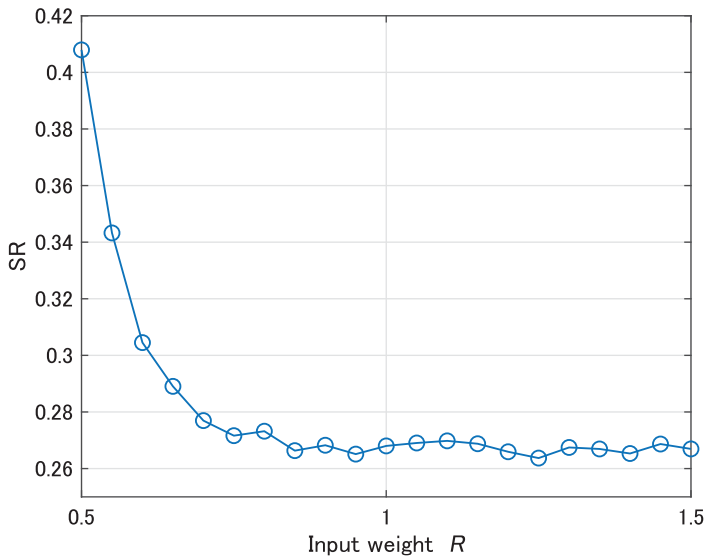


Figure 22. SR of the walking using the NMPC on the uneven terrain no. 1 with respect to the input weight (R).

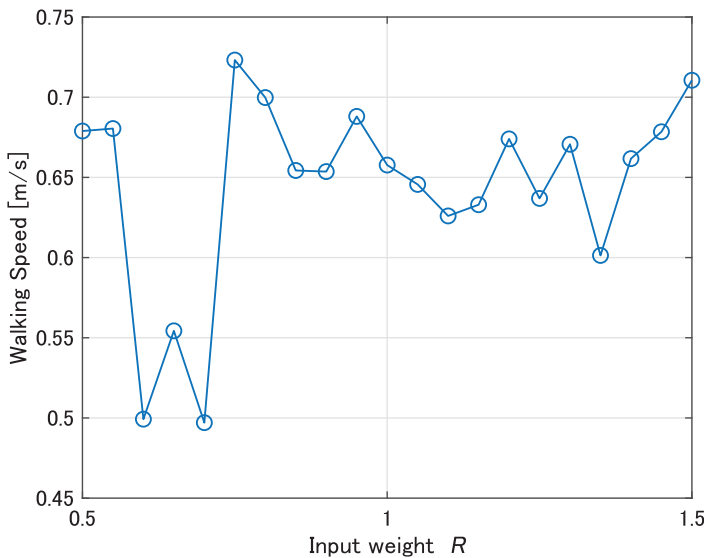


Figure 23. Walking speed using the NMPC on the flat surface with respect to the input weight (R).

performance. In particular, there was a significant difference between walking with and without disturbances. Although this paper was a comparison of two methods, such as NMPC and output-zeroing control, performance comparisons will be conducted with other gait generation methods in the future. Although this study is based on simulation results only, we will verify the practicality of the proposed method by conducting experiments on actual equipment. Additionally, since this study was conducted on the simplest rimless wheel robot, we would further analyze on a walking robot with a more complex mechanism.

Table X. Comparison of the SR values and average walking speeds using the output-zeroing control and NMPC in Unity.

	SR	Speed [m/s]
Output-zeroing control	0.383	0.562
Output-zeroing control in disturbance	0.445	0.534
Proposed NMPC	0.256–0.331	0.497–0.723
Proposed NMPC in disturbance	0.263–0.407	0.604–0.693

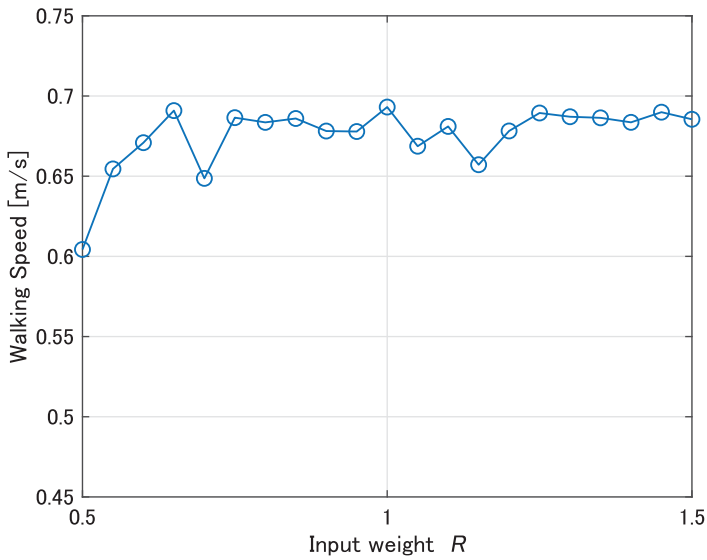


Figure 24. Walking speed using the NMPC on the uneven terrain no. 1 with respect to the input weight (R).

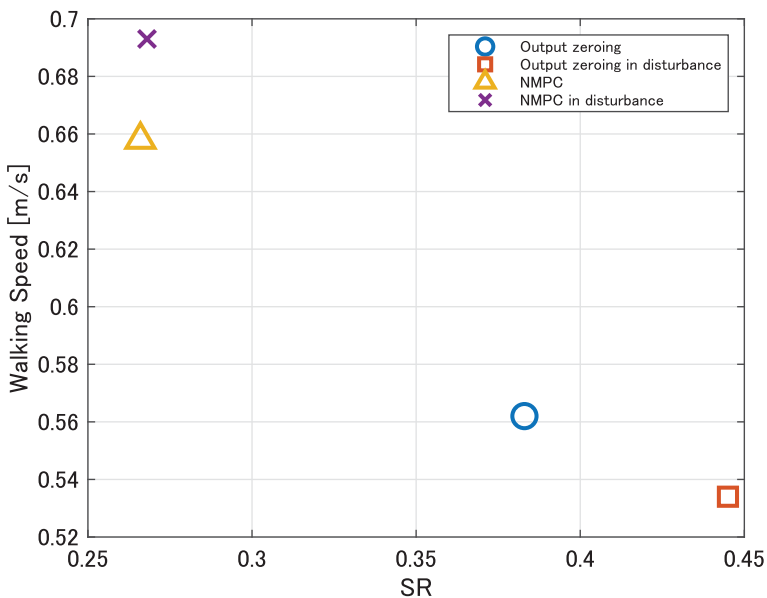


Figure 25. Comparison of the SR values and average walking speeds using the output-zeroing control and NMPC in Unity ($R = 1.0$).

Author contributions. Y. Hanazawa conceived and designed this study. Y. Hanazawa and H. Nishinami verified the usefulness of the proposed method. Y. Hanazawa wrote the article. S. Sagara provided technical support for the study.

Financial support. This research was partially supported by Kitakyushu Foundation for the Advancement of Industry, Science, and Technology.

Competing interests. Authors declare no competing interests exist.

Ethical approval. Not applicable.

Supplementary material. The supplementary material for this article can be found at <https://doi.org/10.1017/S0263574723001522>

References

- [1] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa, “Biped Walking Pattern Generation by Using Preview Control of Zero-Moment Point,” **In: Proc. of IEEE International Conference on Robotics and Automation (ICRA)** (2003) pp. 1620–1626.
- [2] R. Tajima, D. Honda and K. Suga, “Fast Running Experiments Involving a Humanoid Robot,” **In: Proc. of IEEE International Conference on Robotics and Automation (ICRA)** (2009) pp. 1571–1576.
- [3] K. Nishiwaki and S. Kagami, “Strategies for Adjusting the ZMP Reference Trajectory for Maintaining Balance in Humanoid Walking,” **In: Proc. of IEEE International Conference on Robotics and Automation (ICRA)** (2010) pp. 4230–4236.
- [4] S. Satoshi, A. Goswami and P. Vadakkepat, “ASIMO and Humanoid Robot Research at Honda,” **In: Humanoid Robotics: A Reference** (2018) pp. 55–90.
- [5] T. McGeer, “Passive dynamic walking,” *Int. J. Robot. Res.* **9**(2), 62–82 (1990).
- [6] S. H. Collins, A. Ruina, R. Tedrake and M. Wisse, “Efficient bipedal robots based on passive-dynamic walkers,” *Science* **307**(5712), 1082–1085 (2005).
- [7] D. G. E. Hobbelen and M. Wisse, “Limit Cycle Walking,” **In: Humanoid Robots, Human-Like Machines**, chapter **14** (Itech, Vienna, 2007).
- [8] J. W. Grizzle, G. Abba and F. Plestan, “Asymptotically stable walking for biped robots: Analysis via systems with impulse effects,” *IEEE Trans. Autom. Control* **46**(1), 51–64 (2001).
- [9] T. Shimizu, S. Nakaura and M. Sampei, “The Control of a Bipedal Running Robot Based on Output Zeroing Considered Rotation of the Ankle Joint,” **In: Proc. of the 45th IEEE Conference on Decision and Control** (2006) pp. 6456–6461.
- [10] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi and B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion* (CRC press, Boca Raton, FL, 2018).
- [11] Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J. K. Huang and J. W. Grizzle, “Feedback Control of a Cassie Bipedal Robot: Walking, Standing, and Riding a Segway,” **In: Proc. of American Control Conference (ACC)** (2019) pp. 4559–4566.
- [12] T. Ohtsuka, “A continuation/GMRES method for fast computation of nonlinear receding horizon control,” *Automatica* **40**(4), 563–574 (2004).
- [13] Y. Onodera and M. Yamakita, “An Extension of Nonlinear Receding Horizon Control for Switched System with State Jump,” **In: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)** (2005) pp. 984–989.
- [14] A. Murayama and M. Yamakita, “Speed Control of Vehicles with Variable Valve Lift Engine by Nonlinear MPC,” **In: Proc. SICE Annual Conference** (2009) pp. 4128–4133.
- [15] K. Odachi, H. Oyama and M. Yamakita, “Nonlinear Model Predictive Control of SI Engine Using Discrete Time Identification Model,” **In: Proc. SICE Annual Conference** (2015) pp. 1305–1310.
- [16] H. Deng and T. Ohtsuka, “A parallel Newton-type method for nonlinear model predictive control,” *Automatica* **109**, 108560 (2019). <https://www.sciencedirect.com/science/article/pii/S0005109819304212>
- [17] S. Katayama, M. Doi and T. Ohtsuka, “A moving switching sequence approach for nonlinear model predictive control of switched systems with state-dependent switches and state jumps,” *Int. J. Robust Nonlinear Control* **30**(2), 719–740 (2020).
- [18] Unity-Website. <https://www.unity.com>
- [19] F. Asano and J. Kawamoto, “Passive Dynamic Walking of Viscoelastic-Legged Rimless Wheel,” **In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)** (2012) pp. 157–162.
- [20] Y. Hanazawa, “Development of Rimless Wheel with Controlled Wobbling Mass,” **In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)** (2018) pp. 4333–4339.
- [21] S. Sanchez and P. A. Bhounsule, “Design, modeling, and control of a differential drive rimless wheel that can move straight and turn,” *Automation* **2**(3), 98–115 (2021).
- [22] Y. Hanazawa, H. Nishinami and S. Sagara, “Walking experiments of small and lightweight rimless wheel robot,” *Artif. Life Robot.* **27**(4), 706–713 (2022).

- [23] F. Asano and Z. W. Luo, “Asymptotically stable biped gait generation based on stability principle of rimless wheel,” *Robotica* **27**(6), 949–958 (2009).
- [24] F. Asano and M. Suguro, “Limit cycle walking, running, and skipping of telescopic-legged rimless wheel,” *Robotica* **30**(6), 989–1003 (2012).